# Geometric Prior Guided Feature Representation Learning for Long-Tailed Classification

Yanbiao Ma[1*], Licheng Jiao[1], Fang Liu[1], Shuyuan Yang[1], Xu Liu[1] and Puhua chen[1]

[1*]School of Artificial Intelligence, Xidian University, Xi'an, 710071, China.

*Corresponding author(s). E-mail(s): ybmamail@stu.xidian.edu.cn;
Contributing authors: lchjiao@mail.xidian.edu.cn; f63liu@163.com; syyang@xidian.edu.cn; xuliu361@163.com; phchen@xidian.edu.cn;



**Abstract**

Real-world data are long-tailed, the lack of tail samples leads to a significant limitation in the generalization ability of the model. Although numerous approaches of class re-balancing perform well for moderate class imbalance problems, additional knowledge needs to be introduced to help the tail class recover the underlying true distribution when the observed distribution from a few tail samples does not represent its true distribution properly, thus allowing the model to learn valuable information outside the observed domain. In this work, we propose to leverage the geometric information of the feature distribution of the well-represented head class to guide the model to learn the underlying distribution of the tail class. Specifically, we first systematically define the geometry of the feature distribution and the similarity measures between the geometries, and discover four phenomena regarding the relationship between the geometries of different feature distributions. Then, based on four phenomena, feature uncertainty representation is proposed to perturb the tail features by utilizing the geometry of the head class feature distribution. It aims to make the perturbed features cover the underlying distribution of the tail class as much as possible, thus improving the model's generalization performance in the test domain. Finally, we design a three-stage training scheme enabling feature uncertainty modeling to be successfully applied. Experiments on CIFAR-10/100-LT, ImageNet-LT, and iNaturalist2018 show that our proposed approach outperforms other similar methods on most metrics. In addition, the experimental phenomena we discovered are able to provide new perspectives and theoretical foundations for subsequent studies. The code will be available at https://github.com/mayanbiao1234/Geometric-Prior

**Keywords:** Long-Tailed Classification, Representational learning, Geometric prior knowledge

## 1 Introduction

Deep learning has made significant progress in image classification, image segmentation, and other fields benefiting from artificially annotated large-scale datasets. However, real-world data tends to follow a long-tailed distribution [33], and unbalanced classes introduce bias into machine learning models. Numerous approaches have been proposed to mitigate the model bias, such as class re-balancing [6, 12, 50], information augmentation [4, 20, 44, 49] and network structure design [20, 45, 50]. However, the above approach does not work effectively in all cases, and the generalization ability of the model will be greatly limited when the samples of the tail class do not accurately represent its true distribution. We discuss two cases

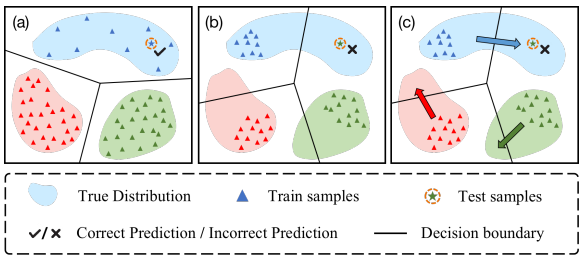


**Fig. 1** (a) When the samples uniformly cover the true data distribution, the model can learn the correct decision boundaries and can correctly classify unfamiliar samples to be tested. (b) When the samples cover only a portion of the true distribution, unfamiliar samples to be tested are highly likely to be misclassified due to the error in the decision boundary. (c) The direction in which the arrow points is the best direction to expand the sample.



of the relationship between the observed and true distributions of the tail classes [5].

- Case 1: The observed samples cover the true data distribution uniformly (As shown in Figure 1a).
- Case 2: The observed samples cover only a small region of the true data distribution (As shown in Figure 2b).

In case 1, although the sample size of the tail classes is small, these samples represent the true data distribution. The main reason for the degradation of the model performance is that at each sampling, samples from the tail class are used with a small probability to calculate loss and update parameters, resulting in inadequate learning of the tail class by the model. Faced with this situation, existing data augmentation methods [4, 41], undersampling [36], oversampling [37, 44], and rebalancing loss [19, 23, 27] can reasonably improve performance. Combining decoupled training with the above approach can improve the performance of the model even further [1, 45]. However, neither rebalancing the sample size nor rebalancing the loss, they fail to increase the information outside the observed training domain. When certain classes are severely underrepresented (i.e., case (2)), these methods have difficulty finding the right direction for adjusting the decision boundaries, so improvements sometimes worsen [5]. Therefore, we pursue to mine knowledge from the well-represented head classes to help recover the true distribution of the tail classes.

In case 2, if the underlying true distribution of the tail classes cannot be recovered, then the model always fails to learn the correct decision boundaries. Even if the model achieves high recognition accuracy in the training set, it still fails to have satisfactory generalization performance when faced with test samples outside the training domain. Therefore, if the direction for recovering the true distribution of tail classes, such as the direction indicated by the three arrows in Figure 1c, can be found, the generalization ability of the model on the tail classes will be significantly improved. It is necessary to explore additional knowledge to guide the tail class to recover the true distribution. However, the head-to-tail knowledge transfer methods [1, 14, 26, 34] currently proposed for the long-tailed challenge do not yet address this issue, so recovering the underlying true distribution using few samples is a meaningful and extremely difficult challenge.

It has been shown that the model bias is caused by the classifier and the long-tailed data does not unbalance the feature representation learning [12, 50]. Also considering that the dimensionality of the feature space is smaller than that of the sample space, we focus on recovering the true distribution of tail classes in the feature space. In this work, our main contributions are summarized as follows.

- We systematically define the geometry of the feature distribution and the similarity measure between the geometries (Section 3.1 and 3.2). Based on this, four surprising experimental phenomena are found which can be used to guide and recover the true distribution of the tail classes (Section 3.3). The most important phenomenon is that similar feature distributions have similar geometries and the similarity between the geometries of the feature distributions decreases as the interclass similarity decreases. We introduce a geometric perspective to recover underrepresented class distributions, providing a theoretical and experimental basis for subsequent studies of class imbalance.
- Based on four experimental phenomena, we propose to model the uncertainty representation of the tail features with geometric information from the feature distribution of the head class (Section 4.1). Specifically, instead of treating samples in the feature space as deterministic points, we perturb them to make the model learn information outside the observed domain by taking into account the geometry of the

class to which the samples belong. Our proposed feature uncertainty modeling can effectively alleviate the model bias introduced by under-represented classes and can be easily integrated into existing networks.

- We propose a three-stage training scheme to apply feature uncertainty representation (Section 4.2). The results of the ablation experiments show that compared to decoupled training, the three-stage training scheme improves the tail class performance while reducing the degradation of the head class performance, resulting in more overall performance improvement of the model.
- Experiments on large-scale long-tailed datasets (Section 5) show that our proposed method significantly improves the performance of tail classes and exhibits state-of-the-art results compared to other similar methods.

# 2 Related Work

## 2.1 Class Rebalancing

The extreme imbalance in the number of samples in the long-tail data prevents the classification model from learning the distribution of the tail classes adequately, which leads to poor performance of the model on the tail classes. Therefore, methods to rebalance the number of samples and the losses incurred per class (i.e., resampling and reweighting) are proposed. Resampling methods are divided into oversampling and undersampling [3, 8, 9, 13, 35, 47]. The idea of oversampling is to randomly sample the tail classes to equalize the number of samples and thus optimize the classification boundaries. The undersampling methods balance the number of samples by randomly removing samples from the head classes. For example, [36] finds that training with a balanced subset of a long-tailed dataset is instead better than using the full dataset. In addition, [12, 50] fine-tune the classifier via a resampling strategy in the second phase of decoupled training. [37] continuously adjusts the distribution of resampled samples and the weights of the two-loss terms during training to make the model perform better. [44] employs the model classification loss from an additional balanced validation set to adjust the sampling rate of different classes.

The purpose of reweighting loss is intuitive, and it is proposed to balance the losses incurred by all classes, usually by applying a larger penalty to the tail classes on the objective function (or loss function) [7, 10, 30, 31, 38, 48]. [27] proposes to adjust the loss with the label frequencies to alleviate class bias. [19] not only assigns weights to the loss of each class, but also assigns higher weights to hard samples. Recent studies have shown that the effect of reweighting losses by the inverse of the number of samples is modest [23, 25]. Some methods that produce more "smooth" weights for reweighting perform better, such as taking the square root of the number of samples as the weight [24]. [6] attributes the better performance of this smoother method to the existence of marginal effects. In addition, [1]proposes to learn the classifier with class-balanced loss by adjusting the weight decay and MaxNorm in the second stage.

## 2.2 Stage-wise training

Decoupling [12] first proposes to decouple the learning process on long-tail data into feature learning and classifier learning, and it finds that re-learning the balanced classifier can significantly improve the model performance. Further, BBN [50] combines the two-stage learning into a two-branch model. The two branches of the model share parameters, with one branch learning using the original data and the other learning using the resampled data. [5] decomposes the features into class-generic features and class-specific features, and it expands the tail class data by combining class-generic features of the head class with class-specific features of the tail class. [49] finds that augmenting data with Mixup in the first stage benefits feature learning and does negligible damage to classifiers trained using decoupling. [45] also observes that long-tailed data does not affect feature learning, and it proposes an adaptive calibration function for improving the cross-entropy loss. [11] considers the effect of noisy samples on the tail class and adaptively assigns weights to the tail class samples by meta-learning in the second stage. Different from the above two-stage training process, we propose a three-stage training scheme. The first two stages are indistinguishable from decoupling training, and in the third stage, the classifier parameters are fixed and the feature

extractor is fine-tuned to adapt to the improved classification boundaries.

Different from the above two-stage training process, we propose a three-stage training strategy. The first two stages are indistinguishable from decoupling training, and in the third stage, the classifier parameters are fixed and the feature extractor is fine-tuned to adapt to the improved classification boundaries.

### 2.3 Head-to-tail knowledge transfer

**Head-to-tail knowledge transfer is more relevant to our work than other methods**. [43] and [21] were first proposed in the face recognition field to transfer variance between classes to augment classes with fewer samples. [43] assumes that the feature distributions of each class are multivariate Gaussian, and the feature distributions of the common and under-represented classes have the same variance, the variance of the head class is used to estimate the distribution of the tail class. [21] assumes that the intra-class angle distribution follows a Gaussian distribution, transfers the intra-class angle distribution of features to the tail class, and constructs a "feature cloud" for each feature to extend the distribution of the tail class.

Similar to the adversarial attack, [14] proposes to transform some of the head samples into tail samples through perturbation-based optimization to achieve tail class augmentation. [5] decomposes the features of each class into class-generic features and class-specific features. During training, the class-specific features of the tail class are fused with the generic features of the head class to generate new features to expand the tail class. This idea is similar to the data augmentation in image space, such as Cutmix. [34] dynamically estimates a set of centers for each class, and then calculates the displacement between the head class feature and the corresponding nearest intra-class center. This displacement is used to combine with the tail class centers to generate new features, thereby increasing the feature diversity of the tail class. [20] proposes to transfer the geometric information of the feature distribution boundaries of the head class to the tail class by enhancing the weights of the tail class classifier. The recently proposed CMO [26] considers that the image of the head class has a rich background, so the image of the tail class can be pasted directly onto the background image of the head class to increase the richness of the tail class. This method can be easily combined with other long-tailed recognition methods.

Previous motivations for head-to-tail knowledge transfer were limited to qualitative analysis or conjecture. Distinguishing from the above studies, we pioneered a geometric perspective of head-tail knowledge transfer. We systematically define the geometry of the distribution and its similarity measure and find direct evidence that the geometry of the head class distribution can help the tail class.

## 3 Motivation

We first define a measure of the geometry of the feature distribution, and then propose a similarity measure between the geometries. Finally, across several benchmark data sets, we discovered four experimental phenomena regarding the relationship between geometric information of feature distributions. Inspired by the experimental phenomenon, we propose to utilize the feature distribution of the head class to help the tail class to recover the underlying distribution.

### 3.1 The Geometry of Data Distribution

In the P-dimensional space, given data $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{P\times n}$ that belongs to the same class, the sample covariance matrix of X can be estimated as

$$\Sigma_X = \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} x_i x_i^T] = \frac{1}{n} X X^T \in \mathbb{R}^{P\times P}.$$

If $\Sigma_X = I_P$ and $I_P$ denotes a unit matrix of order $P$, the distribution of $X$ is said to be isotropic, while the opposite is said to be anisotropic. In practice, the data distribution is usually anisotropic. Considering the two-dimensional case, we can find two vectors $\xi_1$ and $\xi_2$, where $\xi_1$ points to the direction with the largest sample variance, and $\xi_2$ points to the direction with the largest variance among the directions orthogonal to $\xi_1$. $\xi_1$ and $\xi_2$ can be used to anchor the geometry of the two-dimensional distribution.In the high-dimensional case, since $\Sigma_X$ is a

real symmetric matrix, any two of its eigenvectors are orthogonal to each other, and $\xi_i$ points to the direction with the $i$-th largest variance. Analogously to the two-dimensional case, we can use all the eigenvectors of $\Sigma_X$ to anchor the geometry of the distribution.

**Definition 1** (**The geometry of data distribution**). Given a P-dimensional sample set $X$ and the corresponding covariance matrix $\Sigma_X$. The eigendecomposition of $\Sigma_X$ yields P eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_P\}$ and the corresponding P-dimensional eigenvectors $[\xi_1, \xi_2, \ldots, \xi_P] \in \mathbb{R}^{P\times P}$. All eigenvectors of $\Sigma_X$ are considered as bones to anchor the geometry of the distribution of $X$, denoted as

$$GD_X(\xi_1, \xi_2, \ldots, \xi_P),$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_P \geq 0$, $\|\xi_i\|_2 = 1, i = 1, 2, \ldots, P$.

## 3.2 Similarity Measure of Geometry

In the P-dimensional space, given two types of data $X_1 = [x_1, \ldots, x_n] \in \mathbb{R}^{P\times n}$ and $X_2 = [x_1, \ldots, x_n] \in \mathbb{R}^{P\times m}$, their sample covariance matrices are estimated as $\Sigma_{X_1} = \frac{1}{n} X_1 X_1^T \in \mathbb{R}^{P\times P}$ and $\Sigma_{X_2} = \frac{1}{m} X_2 X_2^T \in \mathbb{R}^{P\times P}$, respectively. Performing the eigendecomposition on $\Sigma_{X_1}$ and $\Sigma_{X_2}$, the geometry of the distributions $X_1$ and $X_2$ are denoted as $GD_{X_1}(\xi_{X_1}^1, \ldots, \xi_{X_1}^P)$ and $GD_{X_2}(\xi_{X_2}^1, \ldots, \xi_{X_2}^P)$, respectively, where $\xi_{X_1}^i$ and $\xi_{X_2}^j$ $(i, j = 1, 2, ..., P)$ are the eigenvectors of $\Sigma_{X_1}$ and $\Sigma_{X_2}$, respectively.

**Definition 2** (**Similarity metric between geometry**). Given the geometry of two distributions $GD_{X_1}(\xi_{X_2}^1, \ldots, \xi_{X_2}^P)$ and $GD_{X_2}(\xi_{X_2}^1, \ldots, \xi_{X_2}^P)$, their similarity is defined as

$$S(GD_{X_1}, GD_{X_2}) = \sum_{i=1}^{P} \left\langle \xi_{X_1}^i, \xi_{X_2}^i \right\rangle = \sum_{i=1}^{P} {\xi_{X_1}^i}^T \xi_{X_1}^i.$$

The larger $S(GD_{X_1}, GD_{X_2})$, the more similar the geometry of the distributions $X_1$ and $X_2$. The upper and lower bounds of $S(GD_{X_1}, GD_{X_2})$ are

$$0 \leq S(GD_{X_1}, GD_{X_2}) \leq P.$$

When any pair of eigenvectors $\xi_{X_1}^i$ and $\xi_{X_2}^i$ are co-linear, $S(GD_{X_1}, GD_{X_2})$ reaches the upper bound $P$. When any pair of eigenvectors $\xi_{X_1}^i$ and $\xi_{X_2}^i$ are orthogonal, $S(GD_{X_1}, GD_{X_2})$ takes the lower bound value 0. Taking the two-dimensional distribution as an example, since

$$0 \leq {\phi_{R1}}^T \phi_{B1} + {\phi_{R2}}^T \phi_{B2} \leq {\xi_{R1}}^T \xi_{B1} + {\xi_{R2}}^T \xi_{B2} \leq 2,$$

it is clear that the geometry of the two distributions in Figure A1 is more similar compared to the two distributions shown in Figure A2. The details are described in Appendix A.

## 3.3 Four Discoveries about the Geometry of the Feature Distribution

First define the class similarity measure. Then introduce the four phenomena we found and their experimental setup.

**Definition 3** Given a sample set $D_c = \{\ldots, (x_i, y_c), \ldots\}$ of class $c$, the average prediction score $\frac{1}{|D_c|}\sum_i p(y_c \mid x_i, \theta)$ of all samples belonging to class c is calculated using a deep neural network with trained parameters $\theta$, where $|D_c|$ denotes the sample number of class $c$. Define the class

$$h := argmax_{k\neq c}(\frac{1}{|D_c|}\sum_i p(y_c \mid x_i, \theta))_k$$

that is most similar to class $c$, i.e., the class with the largest logit other than class $c$. Further, the similarity ranking can be done based on logit.

We investigated the relationship between class similarity and the geometry similarity of class distributions on two benchmark datasets: Fashion MNIST [39] and CIFAR-10 [15]. ResNet-18 [40] was adopted as the backbone network and various training schemes were applied to make the performance of ResNet-18 on the two datasets comparable to the state-of-the-art results (See Appendix B for details). First, the similarity between all classes on the two datasets is calculated and ranked. Then, we extracted 64-dimensional features of all samples from both datasets using ResNet18 and calculated the geometry of all class feature distributions. Based on this, we summarize further experiments and findings as follows.

### 3.3.1 Phenomenon 1

As shown in Figure 2, features were extracted using trained ResNet-18 on MNIST [16], Fashion

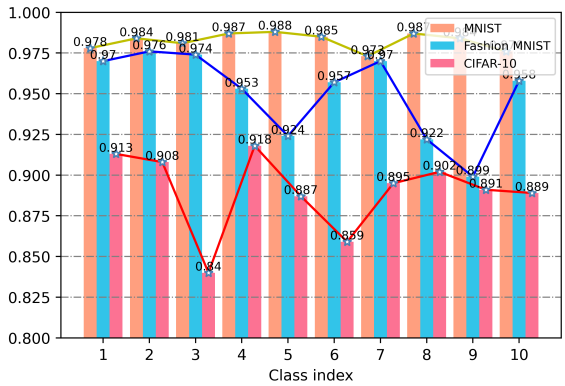


**Fig. 2** The ratio of the sum of the top five eigenvalues to the sum of all eigenvalues after eigendecomposition for the feature embeddings of all classes in the three datasets. The horizontal coordinates are the indexes of the classes, and the specific class names are in Appendix D.

MNIST and CIFAR-10. We find the sum of the eigenvalues corresponding to the first five eigenvectors that are used to represent the geometry of the distribution can reach more than 80% of the sum of all eigenvalues, which means that most of the information of the data distribution can be recovered along the first five eigenvectors.

### 3.3.2 Phenomenon 2

Based on the above observations, we set $P$ in $S(GD_{X_1}, GD_{X_2})$ to 5 and calculate the similarity between geometry of all class feature distributions in Fashion MNIST and CIFAR-10 and plot them in Figure 3a and Figure 3b. We find that **if two classes have high similarity, then the geometry of their feature distributions also exhibit high similarity**. And as the similarity between classes decreases, the similarity between the geometry of the class feature distributions shows a decreasing trend. Take *dog* in CIFAR-10 as an example, its most and least similar classes are *cat* and *automobile*, respectively, and the geometry of the three classes are represented by $GT_{airplane}(\xi_1, \ldots, \xi_{64})$, $GT_{bird}(\eta_1, \ldots, \eta_{64})$ and $GT_{horse}(\zeta_1, \ldots, \zeta_{64})$. Calculate the matrices $M1$ and $M2$ and plot them in Figure 3c and Figure 3d, where $M1_{i,j} = \langle \xi_i, \eta_j \rangle$ and $M2_{i,j} = \langle \xi_i, \zeta_j \rangle$ $(i, j = 1, \ldots, 64)$. It can be observed that $M1$ is closer to a diagonal matrix compared to $M2$, which corresponds to a more similar geometry of *dog* and *cat*.

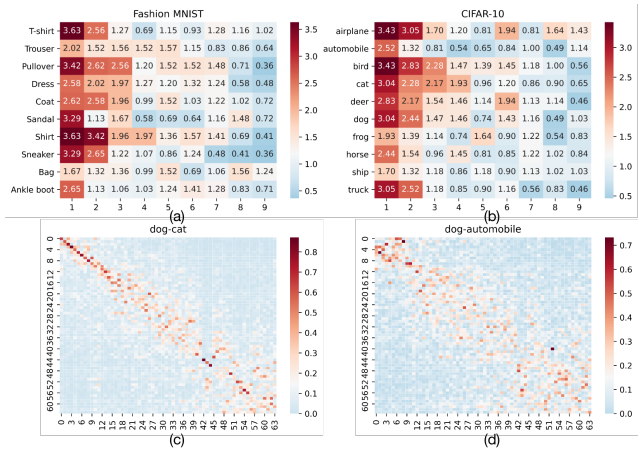


**Fig. 3** (a) The horizontal coordinates are the indexes of the classes, and 1 to 9 indicate the classes that are most similar to the class represented by the vertical coordinates to the least similar, respectively. Each element represents the similarity of the geometry between classes. See Appendix D for detailed class names. (b) Same as (a). (c) The inner product between all eigenvectors of *dog* and all eigenvectors of *cat* in CIFAR-10. The sum of the first five diagonal elements of $M1$ is equal to the value of the element in the first column of the first row in (b). (d) The inner product between all eigenvectors of *dog* and *automobile* in CIFAR-10.

To prove that the above phenomenon does not occur by chance, we give the probability that the experimental results in Figure 3c occur randomly. Given two random vectors in a P-dimensional space, let their inner product be $\delta \in [0, 1]$. The probability density function of $\delta$ is represented as

$$f_P(\delta) = \frac{\Gamma(\frac{P}{2})}{\Gamma(\frac{P-1}{2})\sqrt{\pi}}(1-\delta^2)^{\frac{(P-3)}{2}}.$$

The detailed derivation and proof process of the above equation is shown in the Appendix C. Setting $P$ in $f_p(\delta)$ to 64, when $\delta$ is taken as the first five diagonal elements of $M1$ respectively, the calculation result of $f_P(\cdot)$ is shown in Figure 4a. Considering only the first five diagonal elements of $M1$, the probability of the situation shown in Figure 3c occurring is almost 0. Not only that, we observed numerous such phenomena (see Appendix D), thus implying that the phenomena we found could hardly have occurred by chance.

### 3.3.3 Phenomenon 3

The phenomenon that features distributions of a similar class has similar geometry only occurs

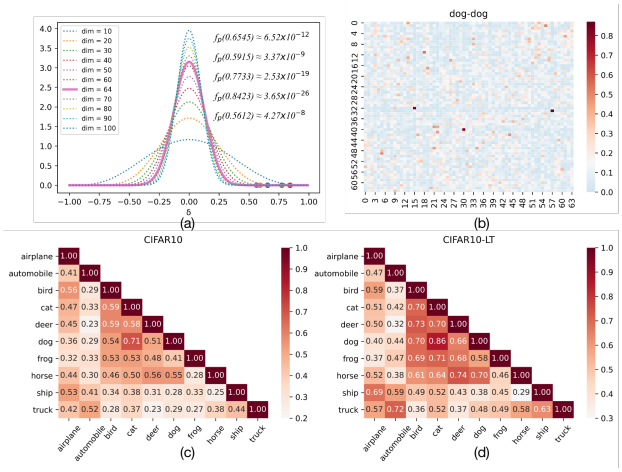


**Fig. 4** (a) The function curve of Equation 1. It can be observed that as the dimensionality increases, any two random vectors tend to be orthogonal to each other. (b) When two different models are used to extract features of *dog* separately, the geometry of the two feature distributions is not similar. (c) Cosine similarity between feature centers of classes on CIFAR-10. (d) Cosine similarity between feature centers of classes on CIFAR-10-LT.

when all features are extracted using the same model. Figure 4b shows that there is a low similarity between the geometry of *dog* computed by two different ResNet18 trained with random initialization. More examples in Appendix E.

#### 3.3.4 Phenomenon 4

We conducted further experiments on CIFAR-10 as well as its long-tailed version CIFAR-10-LT. In CIFAR-10-LT, *airplane*, *automobile*, *bird*, and *cat* are considered head classes and the remaining classes are tail classes. As shown in Figure 4d, we confirmed that the most similar class to the tail class usually belongs to the head class [5] and found that if a tail class and a head class show high similarity in CIFAR-10-LT, they also show high similarity on CIFAR-10.

#### 3.3.5 Summary and inspiration

Combining the above four phenomena, we propose the following idea: the most similar head class is selected for each tail class in the training process, and the geometry of the head class feature distribution is taken as a priori knowledge to guide and recover the underlying true distribution of the tail class.

## 4 Methodology

We first introduce how to leverage the geometric information of the head class feature distribution to model the uncertainty representation of the tail class features, allowing the model to learn the underlying true distribution of the tail class. Then a three-stage training scheme is proposed to apply the feature uncertainty representation.

### 4.1 Feature Uncertainty Representation

Given a tail class $t$, the head class that is most similar to class $t$ is assumed to be $h$. The P-dimensional feature embedding belonging to tail class $t$ is $z_t = \left[z_t^1, \ldots, z_t^{N_t}\right]^T \in \mathbb{R}^{P \times N_t}$ and the feature embedding belonging to head class $h$ is $z_h = \left[z_h^1, \ldots, z_h^{N_h}\right]^T \in \mathbb{R}^{P \times N_h}$, where $N_t$ and $N_h$ denote the sample numbers of class $t$ and class $h$, respectively. The $i$-th feature embedding of $z_t$ is denoted by $z_t^i$. For the model to learn the underlying distribution of the tail class $t$, we want to utilize the existing feature embeddings to generate feature embeddings that can cover the underlying distribution of the tail class $t$. We therefore propose to model the uncertainty representation of $z_t^i$ with the geometry of the feature distribution of class $h$, i.e., $z_t^i$ is no longer considered a deterministic point.

The sample covariance matrix of class $h$ is estimated as $\Sigma_h = \frac{1}{N_h} z_h z_h^T \in \mathbb{R}^{P \times P}$. The eigenvalues of the matrix $\Sigma_h$ are denoted as $\left[\lambda_h^1, \ldots, \lambda_h^P\right] \in \mathbb{R}^P$, where $\lambda_h^1 \geq \cdots \geq \lambda_h^P$. The eigenvector $\left[\xi_h^1, \ldots, \xi_h^P\right] \in \mathbb{R}^{P \times P}$, which corresponds one-to-one with the eigenvalues, anchors the geometry of the class $h$ feature distribution, where $\left\|\xi_h^i\right\|_2 = 1$, $i = 1, \ldots, P$. Since the distributions of similar class have similar geometry, we propose to represent the uncertainty of $z_t^i$ by centering a single feature embedding $z_t^i$ of the tail class $t$ and performing a random translation to $z_t^i$ along a random linear combination of $\xi_h^1, \ldots, \xi_h^P$. Considering that the "scope" of the distribution is larger in the direction with larger eigenvalues [51], an additional weight $\lambda_h^i$ is assigned to $\xi_h^i$ $(i = 1, \ldots, P)$ when the eigenvectors are randomly combined, which means that $z_t^i$ is translated farther with higher probability in the direction with larger eigenvalues. In summary, the final form of

the proposed method can be represented as

$$FUR(z_t^i) = \overbrace{z_t^i + \sum_{j=1}^{P} \epsilon_j \lambda_h^j \xi_h^j}^{Uncertainty\ representation\ of\ z_t^i} \in \mathbb{R}^P$$
$$\epsilon_j \sim N(0,1), j = 1, \ldots, P. \quad (1)$$

$\epsilon_1, \ldots, \epsilon_P$ all follow the standard Gaussian distribution and are independent of each other, and sampling them randomly multiple times can produce new feature embeddings with different translation directions and distances. In particular, when $\epsilon_1 = 1, \epsilon_2, \ldots, \epsilon_P = 0$, $z_t^i$ is translated along $\xi_h^1$ by a distance $\lambda_h^1$. And so on, the maximum translation distances of $z_t^i$ in the direction represented by each feature vector individually are $\lambda_h^1, \ldots, \lambda_h^P$, respectively.

Our proposed method can be integrated as a flexible module after the feature sub-network. It generates augmented samples of tail classes in the feature space to cover the underlying distribution, giving the model better generalization ability on long-tailed data. **Note that this module is only applied during model training and can be discarded during testing without affecting the inference speed**.

### 4.2 Training Scheme

We propose a three-stage training scheme to apply feature uncertainty representation so that the model learns information outside the observed domain. Decoupled training is adopted for the first two phases. In Phase 1, the long-tailed dataset is used to learn the feature sub-network and classifier. In Phase 2, the uncertainty representation of the tail feature is applied to generate new samples for reshaping the decision boundaries. Unlike decoupled training, we additionally add Phase 3 to fine-tune the feature sub-network to adapt it to the new decision boundaries.

- **Phase-1: Initialization training.** Represent an end-to-end deep neural network as a combination of a feature sub-network and a classifier: $M = \{f(x, \theta_1), g(z, \theta_2)\}$, where $\theta_1$ and $\theta_2$ are the parameters of the network. We utilize all images from the dataset to learn the feature sub-network $f(x, \theta_1)$ as well as the classifier $g(z, \theta_2)$. After training is completed, the head classes that are most similar to each tail class are selected based on the average prediction score of the model (see Definition 3), and the geometry of the feature distribution of these head classes is represented by the eigenvectors of the covariance matrix, which will be applied to guide the recovery of the tail class distribution.
- **Phase-2: Reshaping decision boundaries.** Freeze the parameters of $f(x, \theta_1)$ and employ feature uncertainty representation in feature space for the tail class to fine-tune the classifier to improve the performance of the tail class. Specifically, in each iteration, we randomly sample $N_T$ images from the tail class, and then generate $N_A$ augmented samples for each true sample by feature uncertainty representation. Meanwhile, to balance the sample distribution, we directly sample $N_T(1 + N_A)$ samples randomly from the head class. The tail class samples and the head class samples together form a mini-batch containing $2N_T(1+N_A)$ samples for fine-tuning the classifier. The $N_A$ and $N_T$ settings are related to the batch size, and they are described in detail in Section 5.2.
- **Phase-3: Fine-tuned feature sub-network.** Fine-tuning the decision boundary can improve the performance of the tail class while compromising the performance of the head class [43]. This is because the feature sub-network is not well adapted to the new decision boundary. Therefore, we propose to freeze the parameters of $g(z, \theta_2)$ at Phase-3 and fine-tune $f(x, \theta_1)$ with the original long-tailed data.

The above three-phase training process is summarized in Algorithm 1.

## 5 Experiments

### 5.1 Datasets and Evaluation Metrics

We evaluate the effectiveness and generalizability of our approach at CIFAR-10/100-LT [2, 15], ImageNet-LT [22], and iNaturalist2018 [32]. For a fair comparison, the training and test images of all datasets are officially split [42, 46], and the Top-1 accuracy on the test set is utilized as a performance metric.

- Both CIFAR-10 and CIFAR-100 [15] contain $60,000$ images, of which $50,000$ are used for

**Algorithm 1** Feature Uncertainty Representation

**Require**: A long-tailed dataset $D$ containing $S$ samples. A CNN network $M = \{f(x, \theta_1), g(z, \theta_2)\}$, where $\theta_1$ and $\theta_2$ denote the parameters of the feature sub-network and classifier, respectively, and x and z denote the input and feature embedding of the model, respectively.

**for** $epoch = 1$ to $m1$ **do**
    Training model $M$ on dataset $D$ without using any class rebalancing method.
**end for**
Using $M$, the head classes that are most similar to each tail class are calculated, and then the sample covariance matrix of these head classes is calculated.
**for** $epoch = m1$ to $m2$ **do**
    Freeze the parameters $\theta_1$ of the feature sub-network.
    **for** $iteration = 0$ to $\frac{S}{batch\ size}$ **do**
        A mini-batch $\{(x_i, y_i)\}_{i=1}^{batch\ size}$ is sampled from $D$, where the sample numbers from the tail class are $N_T$ and the sample numbers from the head class are $N_T(1 + N_A)$.
        Compute the feature embedding:
$z_i = f(x_i, \theta_1), i = 1, \ldots, (2N_T + N_T N_A)$.
        Uncertainty representation of all features from tail classes: $FUR(z_t^i) = z_t^i + \sum_{j=1}^{P} \epsilon_j \lambda_h^j \xi_h^j \in \mathbb{R}^P$, $t \in tail\ class$, $i \in int[0, N_t]$. h denotes the head class most similar to t.
        $\epsilon_j \sim N(0, 1), j = 1, \ldots, P$. $N_A$ augmented features are generated for the true features of each tail class by randomly sampling $N_A$ times of $\epsilon_j(1, \ldots, P)$.
        A mini-batch with a balanced distribution containing $2N_T(1 + N_A)$ samples is obtained.
        Compute the cross-entropy loss $L(g(z_i, \theta_2), y_i)$ and update the parameters of the classifier:
$\theta_2 = \theta_2 - \alpha \nabla_{\theta_2} L\left(g\left(z_i, \theta_2\right), y_i\right)$.
    **end for**
**end for**
**for** $epoch = m2$ to $m3$ **do**
    Freeze the parameter $\theta_2$ of $g(z, \theta_2)$.
    Fine-tuning the parameters of the feature sub-network using the long-tailed dataset D.
**end for**

training and 10,000 for validation, and they contain 10 and 100 classes, respectively. For a fair comparison, we use the long-tailed version of the CIFAR dataset. The imbalance factor (IF) is defined as the value of the number of the most frequent class training samples divided by the number of the least frequent class training samples. The imbalance factors we employ in our experiments are 10, 50, 100, and 200.

- **ImageNet-LT** is an artificially produced unbalanced dataset utilizing its balanced version (ImageNet-LT-2012 [28]). It with an imbalance factor of 256, contains 1000 classes totaling $115.8k$ images, with a maximum of 1280 images and a minimum of 5 images per class.
- The **iNaturalist2018** dataset is a large-scale real-world dataset that exhibits a long tail. It contains 437,513 training samples from 8,142 classes with an imbalance factor of 500 and three validation samples per class.

## 5.2 Implementation Details

Following the accepted settings [6, 45, 49], the batch sizes on ImageNet-LT and iNaturalist2018 were taken to be 256 and 512, respectively. For a fair comparison, we are consistent with OFA [5] and take $N_A$ to be 3, so $N_T$ is 32 and 64 on ImageNet-LT and iNaturalist2018, respectively. More details of the experimental setup are listed in Table 2. We trained models on CIFAR-10-LT and CIFAR-100-LT using a single NVIDIA 2080Ti GPU and on ImageNet-LT and iNaturalist2018 using 4 NVIDIA 2080Ti GPUs.

## 5.3 Comparative Methods

We train the proposed Feature Uncertainty Representation (FUR) employing decoupled training and three-stage training schemes, respectively. The FUR is then compared with classical and state-of-the-art long-tailed knowledge transfer

**Table 1** Comparison on CIFAR-10-LT and CIFAR-100-LT. The accuracy (%) of Top-1 is reported. The best and second-best results are shown in **underlined bold** and **bold**, respectively. FUR-Decoupled indicates a FUR with decoupled training, and FUR Default indicates a FUR with three-stage training scheme.

| Dataset | Pub. | CIFAR-10-LT | | | | CIFAR-100-LT | | | |
|---|---|---|---|---|---|---|---|---|---|
| Backbone Net | - | ResNet-32 | | | | | | | |
| imbalance factor | - | 200 | 100 | 50 | 10 | 200 | 100 | 50 | 10 |
| Cross Entropy | - | 65.6 | 70.3 | 74.8 | 86.3 | 34.8 | 38.2 | 43.8 | 55.7 |
| BBN [50] | CVPR 2020 | - | 79.8 | 82.1 | 88.3 | - | 42.5 | 47.0 | 59.1 |
| UniMix [41] | NeurIPS 2021 | 78.5 | **82.8** | 84.3 | 89.7 | 42.1 | 45.5 | 51.1 | 61.3 |
| MetaSAug [18] | CVPR 2021 | 76.8 | 80.5 | 84.0 | 89.4 | 39.9 | 46.8 | 51.9 | 61.7 |
| MiSLAS [49] | CVPR 2021 | - | 82.1 | **85.7** | 90.0 | - | 47.0 | 52.3 | 63.2 |
| CDB-W-CE [29] | IJCV 2022 | - | - | - | - | - | 42.6 | - | 58.7 |
| GCL [17] | CVPR 2022 | **79.0** | 82.7 | 85.5 | - | **44.9** | 48.7 | **53.6** | - |
| OFA [5] | ECCV 2020 | 75.5 | 82.0 | 84.4 | **91.2** | 41.4 | 48.5 | 52.1 | **65.3** |
| M2m [14] | CVPR 2020 | - | 78.3 | - | 87.9 | - | 42.9 | - | 58.2 |
| RSG [34] | CVPR 2021 | - | 79.6 | 82.8 | - | - | 44.6 | 48.5 | - |
| CMO [26] | CVPR 2022 | - | - | - | - | - | **50.0** | 53.0 | 60.2 |
| FUR-Decoupled | - | **79.6** | **83.4** | **86.1** | 90.7 | **45.8** | **50.7** | **53.9** | 61.4 |
| FUR | - | **79.8** | **83.7** | **86.2** | **90.9** | **46.2** | **50.9** | **54.1** | **61.8** |

**Table 2** Details of the experimental setup. The 100+50+50 in Epoch indicates the first phase, the second phase, and the third phase are trained for 100, 50, and 50 epochs, respectively.

| Dataset | | CIFAR-10/100-LT | ImageNet-LT | iNaturalist2018 |
|---|---|---|---|---|
| Optimizer: SGD | Mm | 0.9 | | |
| | LR Phase1 | 0.05 | 0.1 | 0.1 |
| | LR Phase2 | 0.001 | 0.001 | 0.001 |
| | LR Phase3 | 0.001 | 0.001 | 0.001 |
| | LR decay | Cosine | Linear | Linear |
| | Batch size | 128 | 256 | 512 |
| | Warm-up | ✓ | ✗ | ✗ |
| Backone | | ResNet-32 | ResNeXt-50 | ResNet-50 |
| Epoch | | 100+50+50 | 100+50+50 | 100+50+50 |

methods, non-transfer data augmentation methods, and other state-of-the-art long-tailed recognition methods. The specific methods are classified as follows.

- **Classical and latest long-tailed knowledge transfer methods**, include OFA [5], M2m [14], RSG [34], GistNet [20], and CMO [26].
- **Other state-of-the-art methods**. They include the two-stage MiSLAS [49], DisAlign [45], BBN [50] with two branches, and the non-transfer augmentation methods UniMix [41], MetaSAug [18], CDB [29] and GCL [17].

## 5.4 Results on CIFAR-10-LT and CIFAR-100-LT

The results on CIFAR-10-LT and CIFAR-100-LT are summarized in Table 1, where our proposed method achieves optimal performance on **six** long-tailed CIFAR datasets and second-best results on the remaining two datasets. Our proposed FUR outperforms GCL by **1%** and **2.2%** on CIFAR-10-LT and CIFAR-100-LT with IF=100, respectively. On the CIFAR-100-LT with IF=10, FUR-Decoupled outperforms the combined CMO by **1.6%**. FUR with a three-stage training scheme outperforms FUR-Decoupled on all datasets, which we will analyze in detail in the next part of the experiment.

Compared to CMO, which randomly pastes the image foreground of the tail class onto the background of the head class image, our proposed FUR relies on the observed prior knowledge to recover the underlying distribution of the tail class. GCL constructs the same "feature cloud" for each feature of the tail class to adjust the model logit, without taking into account the differences in domain characteristics between classes.

**Table 3** Top-1 accuracy (%) of ResNext-50 on ImageNet-LT and Top-1 accuracy (%) of ResNet-50 on iNaturalist2018 for classification. The best and the second-best results are shown in **underline bold** and **bold**, respectively. FUR-Decoupled indicates a FUR with decoupled training, and FUR Default indicates a FUR with three-stage training scheme.

| Methods | Pub. | ImageNet-LT | | | | iNaturalist 2018 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | ResNext-50 | | | | ResNet-50 | | | |
| | | Head | Middle | Tail | Overall | Head | Middle | Tail | Overall |
| BBN [50] | CVPR 2020 | 43.3 | 45.9 | **<u>43.7</u>** | 44.7 | 49.4 | 70.8 | 65.3 | 66.3 |
| DisAlign [45] | CVPR 2021 | 59.9 | 49.9 | 31.8 | 52.9 | 68.0 | 71.3 | 69.4 | 70.2 |
| UniMix [41] | NeurIPS 2021 | - | - | - | 48.4 | - | - | - | 69.2 |
| MetaSAug [18] | CVPR 2021 | - | - | - | 47.3 | - | - | - | 68.7 |
| MiSLAS [49] | CVPR 2021 | **65.3** | **50.6** | 33.0 | 53.4 | **73.2** | **72.4** | 70.4 | 71.6 |
| CDB-W-CE [29] | IJCV 2022 | - | - | - | 38.5 | - | - | - | - |
| GCL [17] | CVPR 2022 | - | - | - | **54.9** | - | - | - | **72.0** |
| OFA [5] | ECCV 2020 | 47.3 | 31.6 | 14.7 | 35.2 | - | - | - | 65.9 |
| RSG [34] | CVPR 2021 | 63.2 | 48.2 | 32.3 | 51.8 | - | - | - | 70.2 |
| GistNet [20] | ICCV 2021 | 52.8 | 39.8 | 21.7 | 42.2 | - | - | - | 70.8 |
| BS + CMO [26] | CVPR 2022 | 62.0 | 49.1 | 36.7 | 52.3 | 68.8 | 70.0 | **72.3** | 70.9 |
| FUR-Decoupled | - | **65.1** | **51.6** | **38.3** | **55.2** | **73.4** | **72.5** | **<u>73.7</u>** | **72.4** |
| FUR | - | **<u>65.4</u>** | **<u>52.2</u>** | 37.8 | **<u>55.5</u>** | **<u>73.6</u>** | **<u>72.9</u>** | **73.1** | **<u>72.6</u>** |

As a result, FUR outperforms similar methods on multiple datasets.

## 5.5 Results on ImageNet-LT and iNaturalist2018

We report in Table 3 not only the overall performance of FUR and FUR-Decoupled on ImageNet-LT and iNaturalist2018 but also additionally add the performance on three subsets of these two datasets, Head (more than 100 images), Middle (20-100 images), and Tail (less than 20 images). Compared to other methods, FUR shows the state-of-the-art overall performance on both ImageNet-LT and iNaturalist2018.

We argue that although the bias of the classifier is mitigated after decoupled training, it is ignored whether the feature sub-network can adapt to the new decision boundaries, which leads to a trade-off in the performance of the head classes. Therefore we add a third stage to fine-tune the feature extractor to adapt it to the latest decision boundaries. FUR-Decoupled outperforms the transfer-based CMO by **3.4%** and **4.8%**, respectively, on the Head subset of the two large-scale long-tailed datasets, benefiting from the fact that FUR relies on prior knowledge rather than randomly recovering the tail class distribution. Although there is a slight degradation in tail

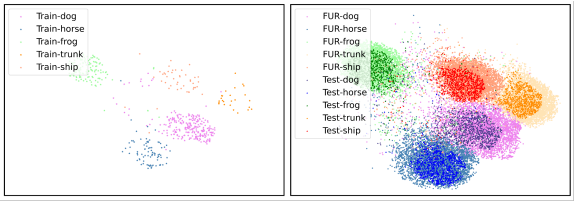


**Fig. 5** Visualization of tail class feature embedding from CIFAR-10-LT with an imbalance factor of 200.

class performance after the third stage, the overall performance of the model and the performance on the head subset are better than the decoupled trained model. Thus both the feature sub-network and the classifier need to be fine-tuned to rebalance the preferences of the model. In addition, the extraordinary performance of FUR-Decoupled on tail classes suggests that our method can recover the underlying distribution of tail classes more efficiently.

## 5.6 Visualization Analysis

To clearly demonstrate that FUR can excel in the recovery of the underlying distribution of tail classes, we visualized the tail features of CIFAR-10-LT via t-SNE. As shown in Figure 5, the training distribution after augmentation with FUR can cover the test distribution well. The above results further show that our proposed method efficiently

recovers the distribution of tail classes. This result further indicates that our proposed method accurately recovers the underlying distribution of the tail classes, allowing the model to perform better on the test set outside the training domain.

## 6 Conclusion

In this work, We discovered four fundamental phenomena regarding the relationship between the geometry of feature distributions, which provide the theoretical and experimental basis for subsequent studies of class imbalance. Inspired by the four phenomena, we propose feature uncertainty representation (FUR) with geometric information for recovering the true distribution of tail classes. After three stages of training, the experimental results show that our proposed method greatly improves the performance of the tail class compared to other methods and ensures the superior performance of the head class at the same time.

## 7 Data Availability Statements

All datasets used in this study are open-access and have been cited in the paper.

## Appendix A

To facilitate the analysis and understanding of the geometry of the feature distributions and the similarity between the geometry, four two-dimensional distributions were generated and plotted in Figure A1 and Figure A2. The geometry of the feature distribution is first introduced. As shown in Figure A1, the direction $\xi_{R1}$ with the largest variance and the direction $\xi_{R2}$ with the largest variance in the direction orthogonal to $\xi_{R1}$ are selected. It can be seen that the geometry and location of the distribution can be anchored by $\xi_{R1}$, $\xi_{R2}$ and the center of the distribution. It is important to note that in this work, we only focus on the shape of the distribution and ignore the location of the distribution. Moreover, if the projection is done along these two directions, the information of the distribution is preserved to the maximum extent.

Observing Figure A1, we can notice that the geometry of the red and blue distributions are more similar because their covariances are similar, i.e., the pattern of variation of the vertical axis with the horizontal axis is similar. The direction of the maximum variance mainly determines the shape of the distribution, and the direction of the second largest variance also plays a role in the shape of the distribution. Obviously, the geometry of the two distributions in Figure A1 is more similar than in Figure A2.

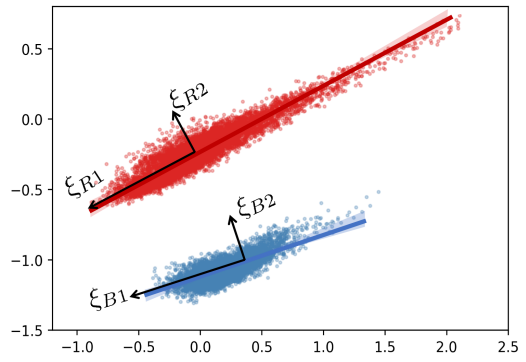


**Fig. A1** Two distributions with similar geometry.

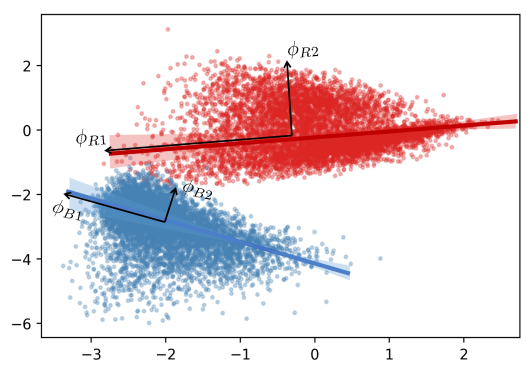


**Fig. A2** Two distributions with low geometry similarity.

## Appendix B

The CIFAR-10 dataset consists of 60,000 images with size $32 \times 32$ from 10 classes, each class contains 6,000 images, of which 5,000 images are used for training and 1,000 images are used for testing. Fashion MNIST has ten classes, each containing 6000 training images and 1000 test images, each with a size of $28 \times 28$.

To improve the generalization ability of the model and prevent the model from overfitting on the training set, we perform three data augmentation operations on the training set: random flip,

random crop, and Cutout. Cutout keeps the model from being overly dependent on certain areas of the image by randomly masking out parts of the image. Considering the size of the image is small, the size 7×7 convolutional kernel of ResNet-18 and the pooling operation tend to lose spatial information, so we remove the maximum pooling layer and adopt the size $3 \times 3$ convolutional kernel instead of the size $7 \times 7$ convolutional kernel.

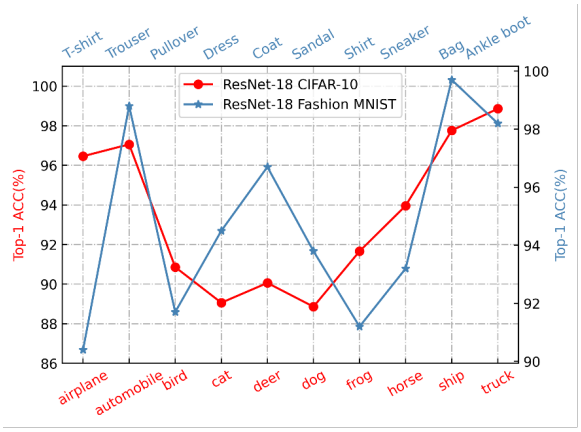


**Fig. B3** Class accuracy of ResNet-18 on Fashion MNIST and CIFAR-10. The class indexes in Figure 2 correspond to the ten numbers of MNIST. For Fashion MNIST and CIFAR-10, the class indexes (i.e., 1 to 10) correspond to the classes from left to right in the above figure, respectively.

We adopt SGD to optimize the model, set the batch size to 128, and the initial learning rate to 0.1. If the loss does not decrease after 10 consecutive epochs, the learning rate becomes 0.5 times of the original, and we train a total of 250 epochs. ResNet-18 achieved an accuracy of 93.46% on CIFAR-10 and 94.82% on Fashion MNIST. The accuracy rates for each class are plotted in Figure B3.

## Appendix C

In the following, we derive the probability density function of the inner product of two random vectors. Without loss of generality, we set x to be a P-dimensional random unit vector and fix y to be a unit vector, i.e.

$$x = (x1, x2, \ldots, x_P), y = (1, 0, ..., 0).$$

The above equation satisfies $x_1^2+x_2^2+\cdots+x_P^2 = 1$. Using spherical transformations

$$\begin{cases} x_1 = r\cos\varphi_1, \\ x_2 = r\sin\varphi_1\cos\varphi_2, \\ x_3 = r\sin\varphi_1\sin\varphi_2\cos\varphi_3, \\ \cdots \\ x_{n-1} = r\sin\varphi_1\sin\varphi_2\ldots\sin\varphi_{n-2}\cos\varphi_{n-1}, \\ x_n = r\sin\varphi_1\sin\varphi_2\ldots\sin\varphi_{n-2}\sin\varphi_{n-1}, \end{cases}$$

where

$$\begin{cases} 0 \le r \le +\infty, \\ 0 \le \varphi_1 \le \pi, \\ \cdots \\ 0 \le \varphi_{n-2} \le \pi, \\ 0 \le \varphi_{n-1} \le 2\pi. \end{cases}$$

The Jacobi determinant of the above transformation is

$$\begin{aligned} J &= \frac{\partial(x_1, x_2, \ldots, x_n)}{\partial(r, \varphi_1, \ldots, \varphi_{n-1})} \\ &= r^{n-1}\sin^{n-2}\varphi_1\sin^{n-3}\varphi_2\ldots\sin\varphi_{n-2}. \end{aligned}$$

Since $x$ is a unit vector, $r = 1$. Notice that $\langle x, y\rangle = x_1 = \cos\varphi_1$, so $\cos\langle x, y\rangle = \cos\varphi_1$. According to the geometric probability,

$$\begin{aligned} &P_n(\varphi_1 \le \theta) = \\ &\frac{\int_0^\theta \sin^{n-2}\varphi_1 d\varphi_1 \cdots \int_0^\pi \sin\varphi_{n-2} d\varphi_{n-2} \int_0^{2\pi} d\varphi_{n-1}}{\int_0^\pi \sin^{n-2}\varphi_1 d\varphi_1 \cdots \int_0^\pi \sin^2\varphi_{n-2} d\varphi_{n-2} \int_0^{2\pi} d\varphi_{n-1}} \\ &= \frac{S_{n-1}\int_0^\theta \sin^{n-2}\varphi_1 d\varphi_1}{S_n}. \end{aligned}$$

Where $S_n$ denotes the surface area of the $n$-dimensional unit sphere. When $k$ is a positive integer,

$$\int_0^\pi \sin^{k-1}\varphi d\varphi = 2\int_0^{\frac{\pi}{2}} \sin^{k-1}\varphi d\varphi,$$

and because

$$\int_0^{\frac{\pi}{2}} \sin^n\varphi d\varphi = \begin{cases} \frac{(2m-1)!!}{(2m)!!}\cdot\frac{\pi}{2}, n = 2m \\ \frac{(2m)!!}{(2m+1)!!}, n = 2m+1 \end{cases},$$

the expression of $S_n$ is obtained. For convenience, we can use the $\Gamma$ function to unify the two, then

$$S_n = \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}.$$

We can obtain

$$P_n(\varphi_1 \leq \theta) = \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2}\sqrt{\pi})} \int_0^{\theta} \sin^{n-2}\varphi_1 d\varphi_1.$$

Further, the probability density function of $\theta$ is calculated as

$$\begin{aligned} f_n(\theta) &= \frac{d}{d\theta} P_n(\varphi_1 \leq \theta) \\ &= \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\sqrt{\pi}} \sin^{n-2}\theta. \end{aligned}$$

We plot the curve of the function $f_n(\theta)$ in Figure C4. It can be seen that the angle between the two random vectors tends to $\pi/2$ as the dimensionality increases.

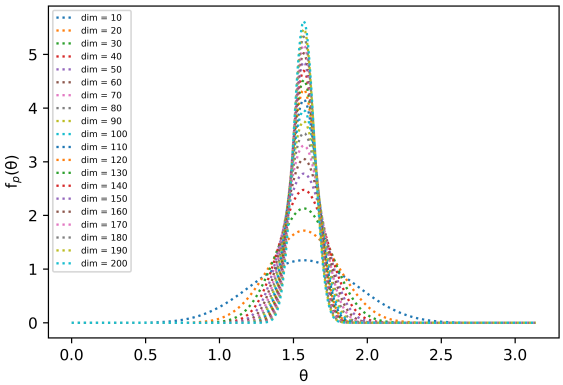


**Fig. C4** The probability density function of the angle between two high-dimensional random vectors.

Let $\delta = \cos\theta$. Then the probability density function of $\delta$ is

$$f_n(\delta) = \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\sqrt{\pi}}(1-\delta^2)^{\frac{(n-3)}{2}}.$$

In the main text, the dimensionality of the feature vector is denoted by P. We plot the curve of the function $f_n(\delta)$ in Figure C5. It can be seen that the inner product between two high-dimensional random vectors tends to 0 as the dimensionality increases, which means that the two random vectors tend to be orthogonal. The above results

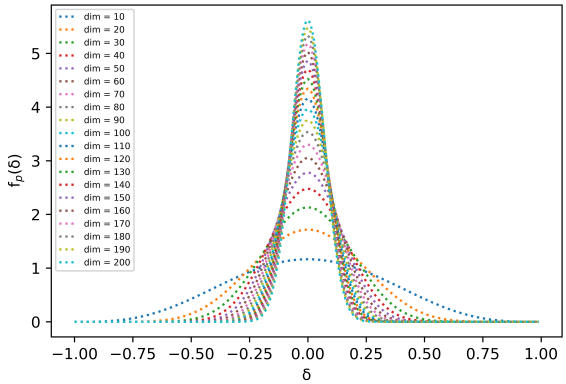


**Fig. C5** The probability density function of the inner product between two high-dimensional random vectors.

prove that our findings did not happen by chance and that the experimental phenomena we summarized are reliable.

# Appendix D

Figure 3 shows the similarity of each class to the other classes on Fashion MNIST and CIFAR-10, and is sorted in descending order of similarity. In Table C1, we list in detail the name of each class in Figure 3a and Figure 3b.

In addition to the geometry similarity between the feature distributions of dog and cat shown in Figure 3c, we also plot the geometry similarity between other classes with high similarity in Figure D6. This evidence strongly suggests that our findings are not accidental.

# Appendix E

In this section, we provide additional experimental results for phenomenon 3. Features of all classes in CIFAR-10 were extracted using two ResNet-18 trained with different initialization parameters, and then the geometry similarity between the feature distributions of the same class extracted by different models was calculated. All additional experimental results are plotted in Figure D6, where it can be observed that the same class of features extracted by the different models does not match phenomenon 2 at all.

**Table C1** Details of all classes in Figure 3a and Figure 3b.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Shirt | Pullover | Dress | Bag | Trouser | Sneaker | Ankle | Sandal | Coat |
| T-shirt | 3.63 | 2.56 | 1.27 | 0.69 | 1.15 | 0.93 | 1.28 | 1.16 | 1.02 |
| | Dress | Coat | Bag | Pullover | Shirt | T-shirt | Ankle | Sneaker | Sandal |
| Trouser | 2.02 | 1.52 | 1.56 | 1.52 | 1.57 | 1.15 | 0.83 | 0.86 | 0.64 |
| | Shirt | Coat | T-shirt | Dress | Trouser | Bag | Sandal | Ankle | Sneaker |
| Pullover | 3.42 | 2.62 | 2.56 | 1.20 | 1.52 | 1.52 | 1.48 | 0.71 | 0.36 |
| | Coat | Trouser | Shirt | T-shirt | Pullover | Bag | Ankle | Sandal | Sneaker |
| Dress | 2.58 | 2.02 | 1.97 | 1.27 | 1.20 | 1.32 | 1.24 | 0.58 | 0.48 |
| | Pullover | Dress | Shirt | Bag | Trouser | Ankle | Sneaker | T-shirt | Sandal |
| Coat | 2.62 | 2.58 | 1.96 | 0.99 | 1.52 | 1.03 | 1.22 | 1.02 | 0.72 |
| | Sneaker | Ankle | Bag | Dress | Shirt | Trouser | T-shirt | Pullover | Coat |
| Sandal | 3.29 | 1.13 | 1.67 | 0.58 | 0.69 | 0.64 | 1.16 | 1.48 | 0.72 |
| | T-shirt | Pullover | Coat | Dress | Bag | Trouser | Ankle | Sandal | Sneaker |
| Shirt | 3.63 | 3.42 | 1.96 | 1.97 | 1.36 | 1.57 | 1.41 | 0.69 | 0.41 |
| | Sandal | Ankle | Coat | T-shirt | Trouser | Bag | Dress | Shirt | Pullover |
| Sneaker | 3.29 | 2.65 | 1.22 | 1.07 | 0.86 | 1.24 | 0.48 | 0.41 | 0.36 |
| | Sandal | Dress | Shirt | Coat | Pullover | T-shirt | Ankle | Trouser | Sneaker |
| Bag | 1.67 | 1.32 | 1.36 | 0.99 | 1.52 | 0.69 | 1.06 | 1.56 | 1.24 |
| | Sneaker | Sandal | Bag | Coat | Dress | Shirt | T-shirt | Trouser | Pullover |
| Ankle | 2.65 | 1.13 | 1.06 | 1.03 | 1.24 | 1.41 | 1.28 | 0.83 | 0.71 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | bird | trunk | ship | cat | horse | deer | automobile | frog | dog |
| airplane | 3.43 | 3.05 | 1.70 | 1.20 | 0.81 | 1.94 | 0.81 | 1.64 | 1.43 |
| | trunk | ship | airplane | forg | cat | horse | bird | dog | deer |
| automobile | 2.52 | 1.32 | 0.81 | 0.54 | 0.65 | 0.84 | 1.00 | 0.49 | 1.14 |
| | airplane | deer | cat | dog | frog | horse | ship | automobile | trunk |
| bird | 3.43 | 2.83 | 2.28 | 1.47 | 1.39 | 1.45 | 1.18 | 1.00 | 0.56 |
| | dog | bird | deer | frog | horse | airplane | ship | trunk | automobile |
| cat | 3.04 | 2.28 | 2.17 | 1.93 | 0.96 | 1.20 | 0.86 | 0.90 | 0.65 |
| | bird | cat | horse | dog | frog | airplane | ship | automobile | truck |
| deer | 2.83 | 2.17 | 1.54 | 1.46 | 1.14 | 1.94 | 1.13 | 1.14 | 0.46 |
| | cat | horse | bird | deer | frog | airplane | truck | ship | automobile |
| dog | 3.04 | 2.44 | 1.47 | 1.46 | 0.74 | 1.43 | 1.16 | 1.03 | 0.49 |
| | cat | bird | deer | dog | airplane | ship | horse | automobile | truck |
| frog | 1.93 | 1.39 | 1.14 | 0.74 | 1.64 | 0.90 | 1.22 | 0.54 | 0.83 |
| | dog | deer | cat | bird | airplane | truck | frog | ship | automobile |
| horse | 2.44 | 1.54 | 0.96 | 1.45 | 0.81 | 0.85 | 1.22 | 1.02 | 0.84 |
| | airplane | automobile | truck | cat | bird | frog | deer | horse | dog |
| ship | 1.70 | 1.32 | 1.18 | 0.86 | 1.18 | 0.90 | 1.13 | 1.02 | 1.03 |
| | airplane | automobile | ship | horse | cat | dog | bird | frog | deer |
| truck | 3.05 | 2.52 | 1.18 | 0.85 | 0.90 | 1.16 | 0.56 | 0.83 | 0.46 |

# References

[1] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6897–6907, 2022.

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information*

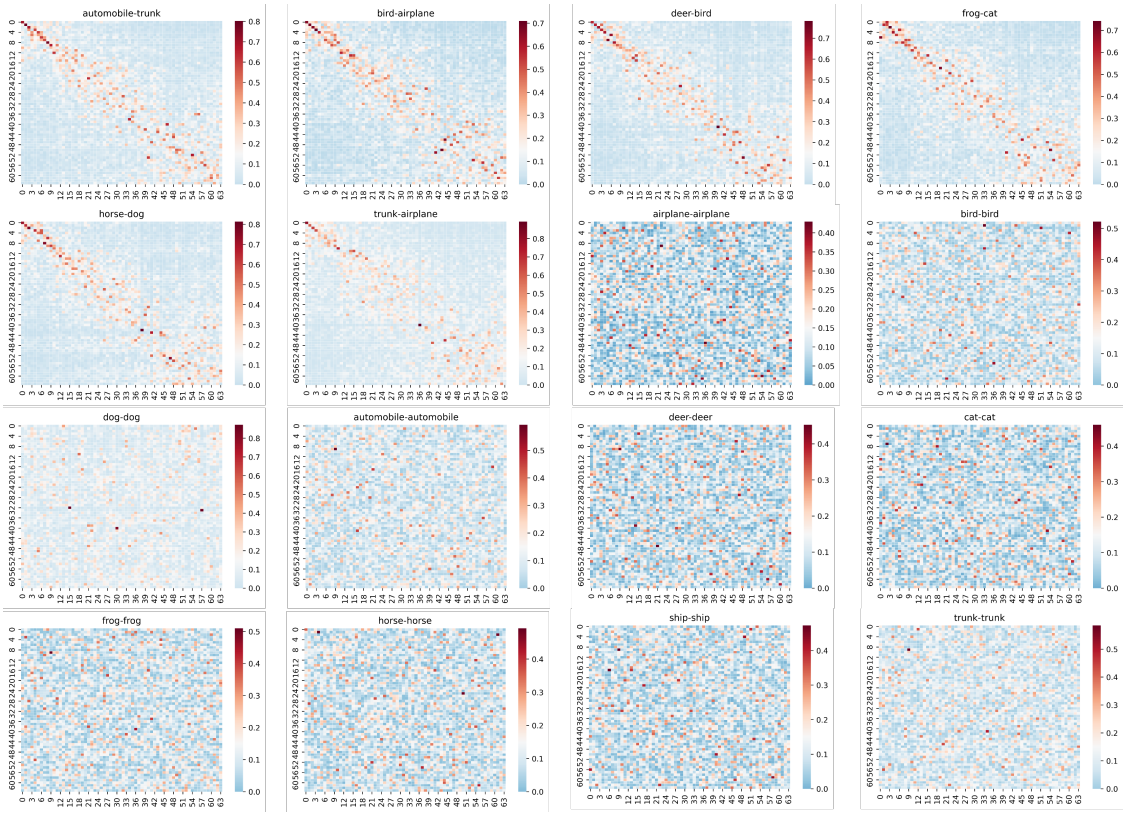


**Fig. D6** Other examples with high geometric similarity in CIFAR-10 and geometry similarity between the same class of feature distributions extracted by different models.

*processing systems*, 32, 2019.

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[4] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. In *European Conference on Computer Vision*, pages 95–110. Springer, 2020.

[5] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020.

[6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[7] Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

[8] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1):18–36, 2004.

[9] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[10] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[11] Shenwang Jiang, Jianan Li, Ying Wang, Bo Huang, Zhang Zhang, and Tingfa Xu. Delving into sample loss curve to embrace noisy and imbalanced data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7024–7032, 2022.

[12] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[13] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[14] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[17] Mengke Li, Yiu-ming Cheung, and Yang Lu. Long-tailed visual recognition via gaussian clouded logit adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6929–6938, 2022.

[18] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5212–5221, 2021.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[20] Bo Liu, Haoxiang Li, Hao Kang, Gang Hua, and Nuno Vasconcelos. Gistnet: a geometric structure transfer network for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8209–8218, 2021.

[21] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2970–2979, 2020.

[22] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[23] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[24] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural*

*information processing systems*, 26, 2013.

[26] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022.

[27] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[29] Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-difficulty based methods for long-tailed visual recognition. *International Journal of Computer Vision*, 130(10):2517–2531, 2022.

[30] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1685–1694, 2021.

[31] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020.

[32] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[33] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

[34] Jianfeng Wang, Thomas Lukasiewicz, Xiaolin Hu, Jianfei Cai, and Zhenghua Xu. Rsg: A simple but effective module for learning imbalanced datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3793, 2021.

[35] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European conference on computer vision*, pages 728–744. Springer, 2020.

[36] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020.

[37] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5017–5026, 2019.

[38] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. *Advances in neural information processing systems*, 30, 2017.

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[40] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[41] Zhengzhuo Xu, Zenghao Chai, and Chun Yuan. Towards calibrated model for long-tailed visual recognition from prior perspective. *Advances in Neural Information Processing Systems*, 34:7139–7152, 2021.

[42] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, pages 1–36, 2022.

[43] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5704–5713, 2019.

[44] Yuhang Zang, Chen Huang, and Chen Change Loy. Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3457–3466, 2021.

[45] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

[46] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

[47] Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 725–734, 2021.

[48] Peilin Zhao, Yifan Zhang, Min Wu, Steven CH Hoi, Mingkui Tan, and Junzhou Huang. Adaptive cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):214–228, 2018.

[49] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021.

[50] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

[51] Yuke Zhu, Yan Bai, and Yichen Wei. Spherical feature transform for deep metric learning. In *European Conference on Computer Vision*, pages 420–436. Springer, 2020.